

UNCLASSIFIED

SAM 86 00310

UNITED STATES AIR FORCE



**ASSISTANT CHIEF OF STAFF,
CENTER FOR
STUDIES AND ANALYSES**

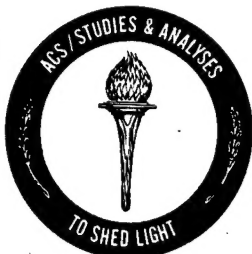
ANALYST'S GUIDE FOR THE STUDIES AND ANALYSES FALLOUT MODEL

SAFALL

Distribution Statement A
Approved for Public Release

19990503 013

MAJOR A.T. HOPKINS
AFCSA/SASM
NOVEMBER 1985



UNCLASSIFIED

24917

SAMI 86 00310

PREFACE

I thank Lt Col Phil Nielsen and Major Bill Davis for helping to define the model output parameters that would be most useful to Air Force personnel in the Attack Information Center. In addition, Bill Davis provided the fallout contour plots (Figures 7 and 10) that he generated from SAFALL output files.

TABLE OF CONTENTS

Preface	i
List of Figures	iii
Abstract	iv
I. Introduction	
General Description	1
Background	1
User Note	2
II. Model Description	
Dose Rate Equation.....	3
Parameters and Algorithms	3
Source Normalization Constant	3
Fission Fraction	4
Cloud Height	4
Arrival Rate Function	4
Crosswind Distribution Function	5
Computation Procedure	5
Dose Rate Options	5
Multiple Burst	7
III. Model Operation	
General	8
Sample Cases	8
IV. References	20
Glossary of Code Variables	21
Appendix. SAFALL FORTRAN Code Listing	23

List of Figures

1. Contour Search Procedure	6
2. SAFALL Option/Data Tree	9
3. SAFALL Worksheet	10
4. SAFALL Scenario Options	11
5. SAFALL Worksheet, Single Burst Contours	12
6. Single Burst Example Session	13
7. Single Burst Example Contours	14
8. SAFALL Worksheet, Multiple Burst Contour	15
9. Multiple Burst Example Session	16
10. Multiple Burst Example Contour	17
11. SAFALL Worksheet, Point Calculation	18
12. Point Calculation Session	19

ABSTRACT

SAFALL, the Studies and Analyses Fallout Model, is a fallout prediction code designed for the Zenith Z-150 microcomputer. The code is intended for use by Air Force personnel in the Attack Information Center. SAFALL produces rapid estimates of radiation dose (rate) from single or multiple nuclear bursts, using an analytic representation of fallout cloud transport and deposition processes. The user supplies input data interactively; output dose (rate) contours or levels can be saved automatically in a user-defined disk file.

I. INTRODUCTION

General Description

The Studies and Analyses Fallout Model, SAFALL, generates rapid predictions of fallout radiation dose (rate) levels that are produced by nuclear bursts on the ground. The model computes first order estimates of dose (rate) from gamma-emitting fallout particles. SAFALL is intended for use by Air Force personnel in the Attack Information Center.

SAFALL was written in FORTRAN77, and compiled with the Microsoft FORTRAN compiler for execution on a Zenith Z-150 microcomputer. The model contains approximately 500 lines of code, using approximately 15K bytes of memory. The executable module occupies approximately 68K bytes of random access memory.

SAFALL prompts the user for the input data and option flags necessary to perform calculations. Output is written to screen, and, if specified, to a user-named disk file.

Background

SAFALL is an analytic simulation of nuclear fallout transport, deposition and irradiation processes. It is a fast-running code based on the theoretical approach of Bridgman and Bigelow (2). Briefly, the model uses a constant, unidirectional wind vector to smear a cloud of falling radioactive particles along the ground downwind of a nuclear burst. The line of peak radioactivity pointing downwind is called the hotline. To speed computations, an analytic expression is used to model the cloud fall process; the model does not perform explicit particle fall calculations.

This type of "smearing" model was selected because it is economical, it fits on the Z-150 and because it closely approximates results of the larger, more sophisticated DoD standard fallout code, DELFIC: Defense Land Fallout Interpretive Code.

SAFALL is analogous to the 26 year old Weapon System Evaluation Group (WSEG) code used for operational fallout studies (5). In fact, SAFALL uses some of WSEG's methods to determine cloud properties. However, SAFALL improves on the

WSEG methodology by incorporating an improved (and user-changeable) particle activity-size spectrum and activity fractionation. Furthermore, SAFALL accounts for the variable settling rates of different-sized fallout particles. SAFALL can also perform multiple burst fallout calculations.

User Note

The following two chapters of this report describe the model's methods and operation. Chapter II presents a brief summary of equations and assumptions. Chapter III explains how to use the model, including three sample runs. A user does not have to wade through the theory presented in chapter II to run SAFALL; go directly to chapter III for operational instructions.

II. MODEL DESCRIPTION

Dose Rate Equation

Fallout dose rates are determined from the following expression for unit time reference dose rate : the hypothetical dose rate received at position (x,y) at one hour after a burst (2).

$$\dot{D}_1(x,y) = k Y ff \frac{g(t_a)}{v_x} f(y,t_a) \quad (1)$$

where

\dot{D}_1 = unit time reference dose rate (rads/hr)
k = source normalization constant (rads/hr)(mi²/kt)
Y = weapon yield (kt)
ff = fission fraction
g(t_a) = arrival rate function (1/hours)
v_x = wind speed (miles/hour)
f(y,t_a) = crosswind distribution function (1/miles)
t_a = time of fallout arrival at (x,y) (hours)

Equation 1 is the basic smearing equation that employs a constant, unidirectional wind, v_x, a normal crosswind distribution function, f(x,t_a), and an analytic arrival rate function, g(t_a). Variable particle fall rates are incorporated in the g(t_a) function.

Parameters and Algorithms

Following are brief descriptions of the key terms and procedures used by SAFALL. A rad (r) is a unit of absorbed dose, equal to 100 ergs of radiation energy deposited in a gram of body tissue.

Source Normalization Constant k. The term k in Equation 1 is the dose rate, at one hour after burst, to a detector centered three feet above a square mile area that has been uniformly covered with radioactive debris from a one kiloton surface burst. This assumes that the fission products' radioactivity level is 530 gamma megacuries per kiloton of fission at one hour with an average gamma ray energy of 0.7 MeV (3). SAFALL uses 2350 (r/hr)(mi²/kt) as derived from analysis of DELFIC code results (2).

Fission Fraction ff. Fission fraction is computed with a bilinear function of weapon yield. High yield thermonuclear weapons generate approximately equal amounts of fission and fusion energy (3). Therefore, fission fraction is set to 0.5 for weapon yields above 1.5 megatons. For lower yields, The fission fraction is represented with a linear function connecting $ff = 0.5$ at 1.5 megatons and $ff = 1$ at 1 kiloton.

$$ff = 1. - YM/3. \quad (2)$$

where

YM = weapon yield in megatons (0.001 to 1.5)

Arrival Rate Function $g(t_a)$. This is the fraction of the cloud's radioactivity arriving on the ground at time, t_a (2).

$$g(t_a) = A(p) \left| \frac{dp}{dt} \right|_{t_a} \quad (3)$$

where

p = particle radius (micrometers)

$A(p)$ = particle activity-size frequency function.

SAFALL uses the DELFIC-default $A(p)$, which is the fractionated sum of two lognormal distributions, derived from nuclear fallout data.

$\frac{dp}{dt}$ = rate at which arriving particle sizes change with time

The arriving particle size (p) and dp/dt depend on the cloud height, particle density and atmospheric properties. Assuming a typical mass density of 2.6 grams/cc and a U.S. Standard Atmosphere, p and dp/dt are obtained from polynomial expansions in cloud height (4).

Cloud Height. The stabilized nuclear cloud is collapsed into a "pancake cloud" at a yield-dependent altitude given by the following equation for cloud height in kilofeet (5).

$$HC = 44. + 6.1 \ln(YM) - .205(\ln(YM)+2.42) |\ln(YM)+2.42| \quad (4)$$

Equation 4 gives the height of the radioactive cloud center above the ground. This height is typically in the bottom third of the visible cloud.

Crosswind Distribution Function $f(y, t_a)$. The crosswind distribution of radioactivity is a Gaussian function (5).

$$f(y, t_a) = \frac{1}{\sqrt{2\pi} \sigma_y(t_a)} \exp \left[-\frac{1}{2} \left(\frac{y}{\sigma_y(t_a)} \right)^2 \right] \quad (5)$$

where

$\sigma_y(t_a)$ = standard deviation (miles)

The standard deviation is the square root of the sum of two squared terms. One term is a characteristic dimension of the cloud at stabilization time. The other term is a time-varying contribution to cloud growth from wind shear. Wind shear is a user input to SAFALL; it can significantly affect the shape of dose (rate) contours. Reference 5 suggests that shear is typically in the range of 0.6 to 2.4 per hour.

Computation Procedure

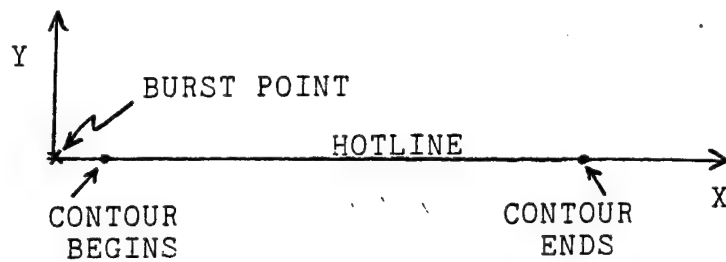
SAFALL determines dose (rate) at a user-specified point or it finds the coordinates of an iso-dose (rate) contour. In the former case, the user selects the point calculation option, specifies (x,y), and the code uses Equation 1 to determine dose rate. In the latter case, the code must search for the set of coordinates that define the contour of a user-specified dose (rate) level. Since the crosswind distribution function is symmetric about the hotline, only half a contour must be located. The x-direction is downwind.

The code finds contour coordinates by first locating the two points on the hotline where the contour begins and ends. It then divides the distance between those two points into 10 segments of equal length. Next, the code searches for non-zero y-values (off-axis distances) of the contour coordinates associated with each of the equally-spaced x-values that separate segments. To speed the process, the search for each successive y-value starts with the previous y-value and marches toward or away from the hotline as required to find where the user-specified dose (rate) occurs. Figure 1 illustrates the procedure.

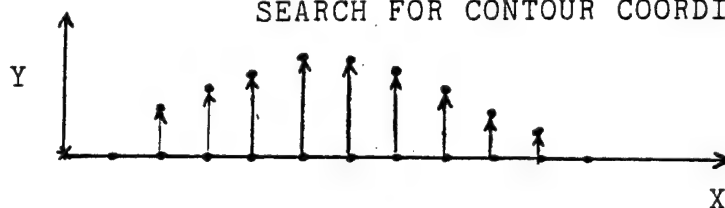
Dose (Rate) Options

SAFALL determines the unit time reference dose rate from Equation 1. However the user can specify options to calculate infinite time dose or dose over a specific time interval. Dose

1. FIND CONTOUR INTERSECTIONS ON HOTLINE



2. DIVIDE HOTLINE INTO 10 SEGMENTS;
SEARCH FOR CONTOUR COORDINATES.



3. REFLECT COORDINATES BELOW HOTLINE; GENERATE CONTOUR.

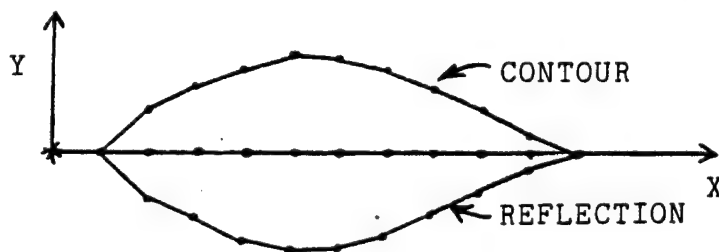


FIGURE 1. CONTOUR SEARCH PROCEDURE

is determined from unit time reference dose rate by assuming that fission products decay with time according to the Way-Wigner approximation (3).

$$D = \int_{t_a}^{t_d} \dot{D}_1 t^{-1.2} dt = 5 \dot{D}_1 (t_a^{-.2} - t_d^{-.2}) \quad (6)$$

where

D = finite time dose received from arrival time (t_a)
to departure time (t_d) (rads)

If the receiver stays for an indefinitely long period, $t_d^{-.2} \rightarrow 0$ and Equation 6 gives the infinite time dose.

Finite time dose calculations permit the user to specify t_a and t_d . The code adjusts integration limits accordingly, warning the user if the receiver departs before the cloud arrives.

Multiple Burst

SAFALL uses the technique described in reference (2) to perform multiple burst calculations. By uniformly distributing the bursts along a crosswind line, the code essentially smears a line source, instead of a point source, along the ground downwind. Total activity is conserved, and crosswind dispersion is represented with a cumulative normal function that sums contributions from each burst (1). The user specifies the number of bursts and the crosswind distance over which they are spread.

III. RUNNING THE MODEL

General

To run SAFALL, load the Microsoft Disk Operating System, then place the SAFALL disk into the default drive. Type SAFALL after the system prompt, then press the return key. This starts an interactive session in which the code prompts the user to enter the data necessary for fallout calculations. Figure 2 illustrates the options and data entry sequence. Figure 3 is a worksheet to help the user to anticipate and gather the information needed to run the code.

Sample Cases

With options for: (i) contour or point calculations, (ii) dose rate, infinite time dose or finite time dose, and (iii) single or multiple burst scenarios, there are twelve different ways that SAFALL can represent nuclear fallout environments. The twelve combinations are shown in Figure 4.

Single Burst Contours. Figure 5 is the worksheet for a 1 megaton burst in a 30 mile per hour wind with 1.15/hour shear. A copy of the interactive session with SAFALL is shown in Figure 6. Contour dimensions illustrated in Figure 7 agree well with results published in Reference 2.

Multiple Burst Contour. Similarly, Figure 8 is the worksheet for twenty three hundred 1 megaton bursts in a 35 mile per hour wind with a shear of 1/hour. A copy of the interactive session is shown in Figure 9. Again, SAFALL results plotted in Figure 10 agree with published contours.

Single Burst Point Calculation. Figure 11 is the worksheet for a 1 megaton burst in a 50 mile per hour wind with 1/hour shear. Finite time dose is computed for a receiver who arrives at a downwind point at 5 hours after burst time and leaves 48 hours later. The point coordinates ($x = 100$ miles, $y = 10$ miles) are measured with the burst point as origin of a Cartesian grid; the x -direction is downwind, and the y -direction is crosswind. Figure 12 is a copy of the interactive session for this run.

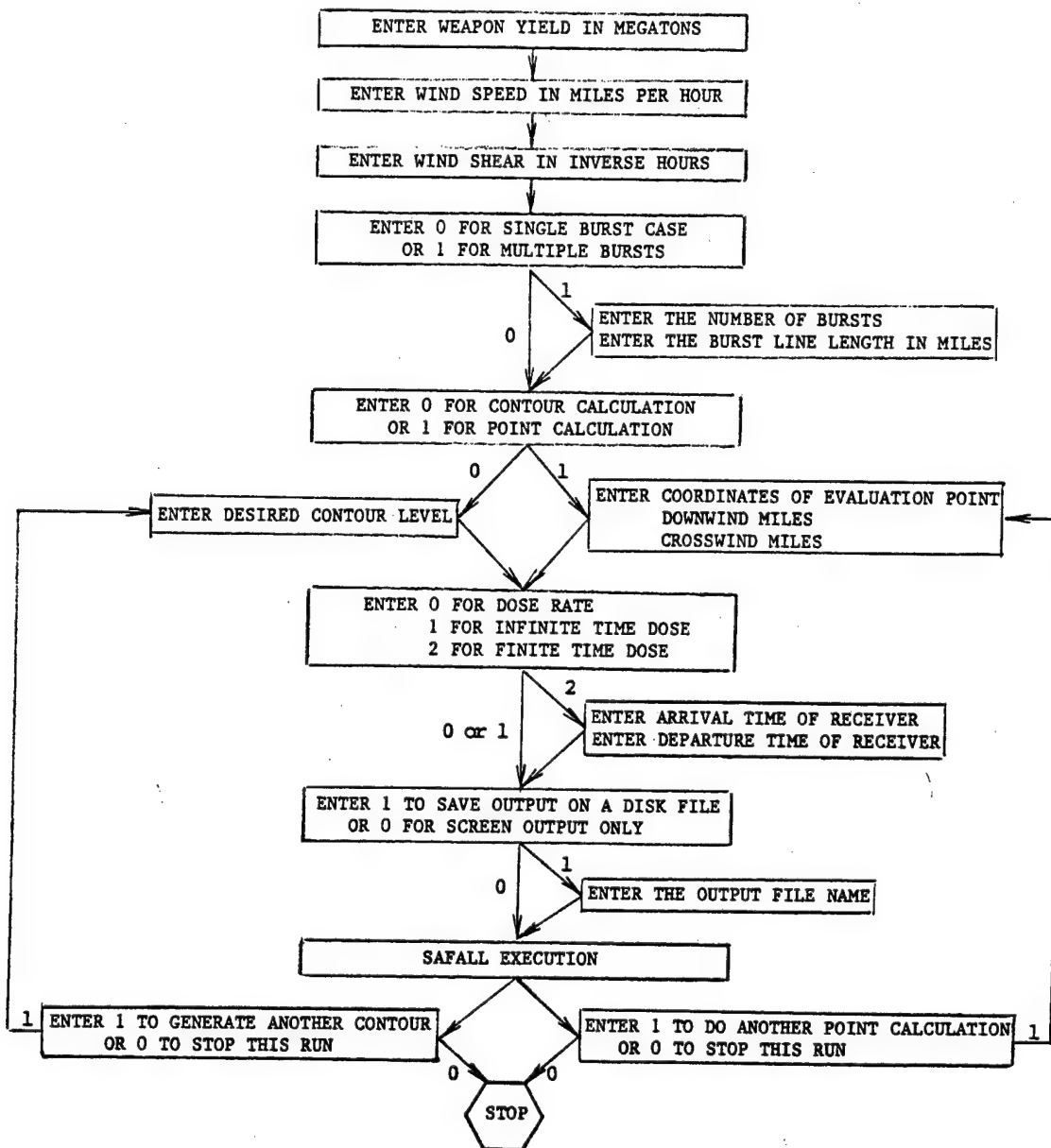


FIGURE 2. SAFALL OPTION/DATA TREE

SAFALL WORKSHEET

ENTRY	UNITS	OPTIONS
Weapon Yield	Megatons	.001-15.
Wind Speed	Miles/Hour	>0
Wind Shear	1/Hours	-
Single Burst		0
or	Flag	or
Multiple Burst		1
If Multiple Burst:		
Number of Bursts	Number	>0
Burst Line Length	Miles	>0
Contour Calculation		0
or	Flag	or
Point Calculation		1
If Contour:		
Contour Level	rads,r/hr	>0
If Point:		
Downwind Coordinate	Miles	>0
Crosswind Coordinate	Miles	-
Dose Rate		0
Infinite Time Dose	Flag	or 1
Finite Time Dose		or 2
If Finite Time Dose:		
Receiver Arrival Time	Hours	>0
Receiver Departure Time	Hours	>0
Save to Disk File:	Flag	0=NO or 1=YES
If Save to Disk:		*
Name of File	Name	d:f.e
Another Calculation?	Flag	0=NO or 1=YES

*

d:f.e = disk drive:file name.extension

FIGURE 3. SAFALL WORKSHEET

SAFALL SCENARIO OPTIONS

	Contour		Point	
	Single	Multiple	Single	Multiple
Dose Rate	1	4	7	10
Infinite Time Dose	2	5	8	11
Finite Time Dose	3	6	9	12

FIGURE 4. SAFALL SCENARIO OPTIONS

SAFALL WORKSHEET

ENTRY	UNITS	OPTIONS	
Weapon Yield	Megatons	.001-15.	1
Wind Speed	Miles/Hour	>0	30
Wind Shear	1/Hours	-	1.15
Single Burst or Multiple Burst	Flag	0 or 1	0
If Multiple Burst: Number of Bursts	Number	>0	
Burst Line Length	Miles	>0	
Contour Calculation or Point Calculation	Flag	0 or 1	0
If Contour: Contour Level	rads,r/hr	>0	100
If Point: Downwind Coordinate	Miles	>0	
Crosswind Coordinate	Miles	-	
Dose Rate		0	
Infinite Time Dose	Flag	or 1	0
Finite Time Dose		or 2	
If Finite Time Dose: Receiver Arrival Time	Hours	>0	
Receiver Departure Time	Hours	>0	
Save to Disk File:	Flag	0=NO or 1=YES	0
If Save to Disk:		*	
Name of File	Name	d:f.e	
Another Calculation?	Flag	0=NO or 1=YES	0

*

d:f.e = disk drive:file name.extension

FIGURE 5. SAFALL WORKSHEET, SINGLE BURST CONTOURS

safall

*** AFCSA/SASM NUCLEAR FALLOUT MODEL ***
Version 4.2

Enter weapon yield in megatons: 1

Enter wind speed in miles per hour: 30

Enter wind shear in inverse hours: 1.15

Enter 0 for single burst case,
or 1 for multi-burst case: 0

Enter 0 for contour calculation,
or 1 for point calculation: 0

Enter desired contour level: 100

Enter 0 for dose rate,
1 for infinite time dose,
or 2 for finite time dose: 0

Enter 1 to save output on a disk file,
or 0 for screen output only: 0

SUMMARY

Contour Position Calculation

Weapon Yield = 1.000 megatons
Wind Speed = 30. mph
Wind Shear = 1.1 per hour
Dose Rate = 100. r/hr

Single Burst Case

Contour intersections with hotline:
upwind .300E+01 miles
downwind .123E+03 miles

x	y	-y	toa
.300E+01	.000E+00	.000E+00	.100E+00
.150E+02	.580E+01	-.580E+01	.500E+00
.270E+02	.660E+01	-.660E+01	.900E+00
.390E+02	.710E+01	-.710E+01	.130E+01
.510E+02	.740E+01	-.740E+01	.170E+01
.630E+02	.750E+01	-.750E+01	.210E+01
.750E+02	.730E+01	-.730E+01	.250E+01
.870E+02	.680E+01	-.680E+01	.290E+01
.990E+02	.600E+01	-.600E+01	.330E+01
.111E+03	.460E+01	-.460E+01	.370E+01
.123E+03	.000E+00	.000E+00	.410E+01

Enter 1 to generate another contour,
or 0 to stop this run: 0
Stop - Program terminated.

FIGURE 6. SINGLE BURST EXAMPLE SESSION

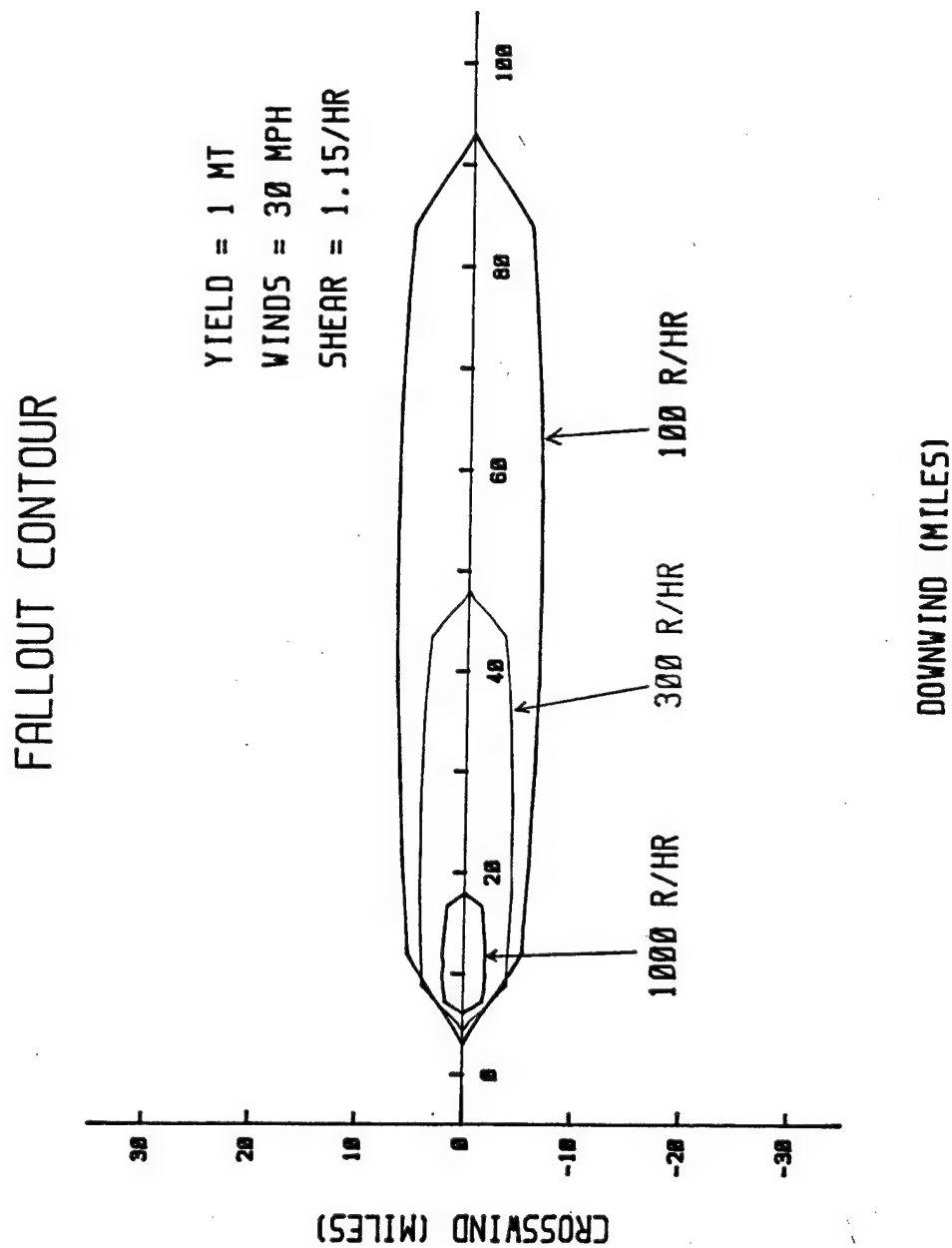


FIGURE 7. SINGLE BURST EXAMPLE CONTOURS

SAFALL WORKSHEET

ENTRY	UNITS	OPTIONS	
Weapon Yield	Megatons	.001-15.	1
Wind Speed	Miles/Hour	>0	35
Wind Shear	1/Hours	-	1
Single Burst or Multiple Burst	Flag	0 or 1	1
If Multiple Burst: Number of Bursts	Number	>0	2300
Burst Line Length	Miles	>0	180
Contour Calculation or Point Calculation	Flag	0 or 1	0
If Contour: Contour Level	rads,r/hr	>0	1500
If Point: Downwind Coordinate	Miles	>0	
Crosswind Coordinate	Miles	-	
Dose Rate		0	
Infinite Time Dose	Flag	or 1	1
Finite Time Dose		or 2	
If Finite Time Dose: Receiver Arrival Time	Hours	>0	
Receiver Departure Time	Hours	>0	
Save to Disk File:	Flag	0=NO or 1=YES	0
If Save to Disk:		*	
Name of File	Name	d:f.e	
Another Calculation?	Flag	0=NO or 1=YES	0

*

d:f.e = disk drive:file name.extension

FIGURE 8. SAFALL WORKSHEET, MULTIPLE BURST CONTOUR

safall

*** AFCSA/SASM NUCLEAR FALLOUT MODEL ***
Version 4.2

Enter weapon yield in megatons: 1
Enter wind speed in miles per hour: 35
Enter wind shear in inverse hours: 1
Enter 0 for single burst case,
or 1 for multi-burst case: 1
Enter the number of bursts: 2300
Enter the burst line length in miles: 180
Enter 0 for contour calculation,
or 1 for point calculation: 0
Enter desired contour level: 1500
Enter 0 for dose rate,
1 for infinite time dose,
or 2 for finite time dose: 1
Enter 1 to save output on a disk file,
or 0 for screen output only: 0

SUMMARY

Contour Position Calculation

Weapon Yield = 1.000 megatons
Wind Speed = 35. mph
Wind Shear = 1.0 per hour
Infinite Time Dose = 1500. r

Multiple Burst Case

2300 bursts along a line 180.0 miles long

Contour intersections with hotline:

upwind .175E+01 miles
downwind .216E+04 miles

x	y	-y	toa
.175E+01	.900E+02	-.900E+02	.500E-01
.217E+03	.109E+03	-.109E+03	.620E+01
.433E+03	.117E+03	-.117E+03	.124E+02
.648E+03	.121E+03	-.121E+03	.185E+02
.863E+03	.122E+03	-.122E+03	.247E+02
.108E+04	.119E+03	-.119E+03	.308E+02
.129E+04	.113E+03	-.113E+03	.370E+02
.151E+04	.103E+03	-.103E+03	.431E+02
.173E+04	.890E+02	-.890E+02	.493E+02
.194E+04	.665E+02	-.665E+02	.554E+02
.216E+04	.000E+00	.000E+00	.616E+02

Enter 1 to generate another contour,
or 0 to stop this run: 0
Stop - Program terminated.

FIGURE 9. MULTIPLE BURST EXAMPLE SESSION

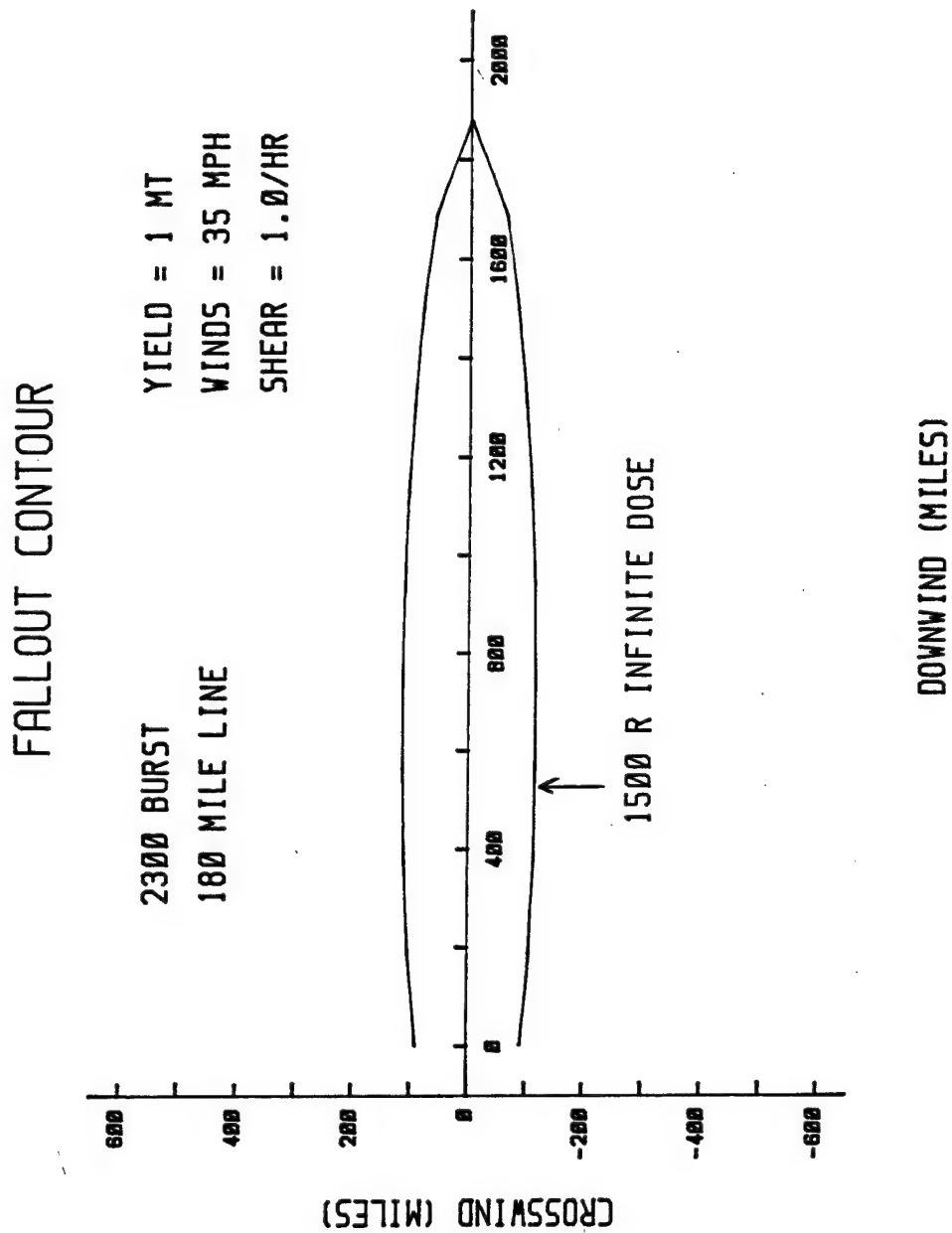


FIGURE 10. MULTIPLE BURST EXAMPLE CONTOUR

SAFALL WORKSHEET

ENTRY	UNITS	OPTIONS	
Weapon Yield	Megatons	.001-15.	1
Wind Speed	Miles/Hour	>0	50
Wind Shear	1/Hours	-	1
Single Burst or Multiple Burst	Flag	0 or 1	0
If Multiple Burst: Number of Bursts	Number	>0	
Burst Line Length	Miles	>0	
Contour Calculation or Point Calculation	Flag	0 or 1	1
If Contour: Contour Level	rads,r/hr	>0	
If Point: Downwind Coordinate	Miles	>0	100
Crosswind Coordinate	Miles	-	10
Dose Rate		0	
Infinite Time Dose	Flag	or 1	2
Finite Time Dose		or 2	
If Finite Time Dose: Receiver Arrival Time	Hours	>0	5
Receiver Departure Time	Hours	>0	10
Save to Disk File:	Flag	0=NO or 1=YES	1
If Save to Disk:		*	
Name of File	Name	d:f.e	TEST
Another Calculation?	Flag	0=NO or 1=YES	0

*
d:f.e = disk drive:file name.extension

FIGURE 11. SAFALL WORKSHEET, POINT CALCULATION

safall

*** AFCSA/SASM NUCLEAR FALLOUT MODEL ***
Version 4.2

Enter weapon yield in megatons: 1

Enter wind speed in miles per hour: 50

Enter wind shear in inverse hours: 1

Enter 0 for single burst case,
or 1 for multi-burst case: 0

Enter 0 for contour calculation,
or 1 for point calculation: 1

Enter coordinates of evaluation point.
downwind miles: 100
crosswind miles: 10

Enter 0 for dose rate,
1 for infinite time dose,
or 2 for finite time dose: 2

Enter arrival time of receiver (hours): 5
Enter departure time of receiver (hours): 48

Enter 1 to save output on a disk file,
or 0 for screen output only: 0

SUMMARY

Point Calculation of Finite Time Dose
Receiver Arrival Time : .500E+01
Receiver Departure Time : .480E+02

Weapon Yield =	1.000 megatons
Wind Speed =	50. mph
Wind Shear =	1.0 per hour

Single Burst Case

xpt = .100E+03 ypt = .100E+02 toa = .200E+01 D = .220E+02

Enter 1 to perform another point calculation,
or 0 to stop this run: 0

Stop - Program terminated.

FIGURE 12. POINT CALCULATION SESSION

IV. REFERENCES

1. Abrahamowitz, Milton and Irene A. Stegun (editors). Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables. Washington DC. National Bureau of Standards, December 1972.
2. Bridgman Charles J. and Winfield S. Bigelow. A New Fallout Prediction Model. Health Physics, 43 (2):205-218 (August 1982).
3. Glasstone, Samuel and Philip J. Dolan. The Effects of Nuclear Weapons. Third Edition. Washington DC. U.S.Government Printing Office, 1977.
4. Hopkins, Arthur T. Working Paper Describing Correlation of Laurent Series Coefficients with Altitude. Air Force Institute of Technology. Wright-Patterson AFB, OH.
5. Pugh, George E. and R.J. Galliano. An Analytic Model of Close-in Deposition of Fallout For Use in Operational Type Studies. Weapon Systems Evaluation Group Memorandum RM10. Washington DC, 1959.

GLOSSARY OF CODE VARIABLES

aa : product of weapon and fallout constants
al : array of two median particle radii
bl : length of burst line
bt : array of two log slopes in particle activity-size spectrum
c : array of seven Laurent series coefficients
cn : cumulative normal function
cnz1 : cn evaluated for argument z1
cnz2 : cn evaluated for argument z2
ddl : dose (rate) value
dd2 : dose (rate) value
deltax : incremental distance in downwind direction
dn : time step increment
drc : user-defined contour level
drdt : time rate of change in particle size arriving on ground
drmax : maximum dose (rate) on hotline
drxy : dose (rate) at (x,y)
dul : dose (rate) on hotline upwind of drmax
du2 : dose (rate) on hotline upwind of drmax
dy : incremental distance in crosswind direction
dl-d7 : seven arrays of seven polynomials each: used to
generate Laurent series coefficients (c)
fdum : dummy variable for fxy
ff : fission fraction
fname : name of output disk file
fr : array of two fractionation ratios
fxy : crosswind distribution function
g : activity arrival rate function
gn : variable to increment time steps
gtalk : character variable for Graftalk counter on disk output
hk : hc in kilometers
inf : flag for infinite time dose calculation
ls : integer remainder for Graftalk counter
mb : flag for multiple burst calculation
more : flag to signal another code run
ms : integer for Graftalk counter
nb : number of bursts
nc : character array for Graftalk counter
ncase : flag for Graftalk counter on disk output
ncon : flag to signal point or contour calculation
ncpr : character array with integers for Graftalk counter
ndisk : flag to enable output to be written to disk
nfl : flag to signal end of iterations
nocon : flag to signal existence of drc
p : argument of log normal distribution function
r : particle radius
ro : median radius in log normal particle number-size
distribution function

shry : crosswind shear
 sigh : deviation of cloud thickness
 sigo : deviation of stabilized cloud width
 snc : source normalization constant
 sigy : deviation of cloud width at arrival time
 t : elapsed time since burst
 tc : time constant for toroidal growth
 tdn : cloud arrival time at xdn
 tin : arrival time of receiver for finite time calculation
 tmax : time at which particles land at maximum dose rate point
 toa : time of cloud arrival at point (x,y)
 tout : departure time of receiver for finite time calculation
 ts : time limit for toroidal growth
 tup : cloud arrival time at xup
 tz : time increment for g(t) calculation
 vx : effective wind speed
 x : downwind coordinate
 xdn : downwind hotline coordinate of drc
 xmax : hotline coordinate of maximum dose rate
 xpt : downwind coordinate for point calculation
 xup : upwind hotline coordinate of drc
 y : crosswind coordinate
 ym : weapon yield in megatons
 yold : y-value from previous iteration
 ypt : crosswind coordinate for point calculation
 z1 : argument for cumulative normal function
 z2 : argument for cumulative normal function

APPENDIX

SAFALL FORTRAN Code Listing

```

program safall
dimension d1(7),d2(7),d3(7),d4(7),d5(7),d6(7),d7(7)
common /p/ al(2),bt(2),fr(2),ro,c(7)
common /y/ ym,ff
common /c/ drc
common /s/ snc
common /v/ vx,shry
common /t/ tc,sigo,sigh
common /m/ mb,nb,bl
common /xy/ tmax,xmax,drmax,xup,xdn,du2,dd2,nocon
common /inf/ inf,tin,tout
common /nd/ ndisk,ncon,ncase
character*15 fname

c
nocon=0
do 10 i=1,7
10 c(i)=0.

c
data d1/ .1568856848e-07,-.1818243397e-07, .5203039670e-08,
c      -.5693308769e-09, .1994484257e-10,-.2735221932e-12,
c      .1259851522e-14/
data d2/- .4262994825e-07, .6693437385e-07,-.2722740725e-07,
c      .4757331620e-08,-.1955377245e-09, .3030762717e-11,
c      -.1611815768e-13/
data d3/- .1415016611e-06, .1536130380e-06,-.4226855150e-07,
c      -.7623625786e-08, .4437252345e-09,-.7725847819e-11,
c      .4366090399e-13/
data d4/- .4385983125e-07, .4344379219e-07, .5776466842e-06,
c      -.2324699454e-07, .3288743990e-09,-.2012703249e-11,
c      .7274195950e-14/
data d5/ .2489431690e-06, .3873893848e-05,-.9536704134e-08,
c      -.8350691547e-08, .4145952544e-09,-.6988253916e-11,
c      .3921197877e-13/
data d6/- .1777248586e-05,-.1200921373e-08, .2176153448e-08,
c      -.2019724989e-09, .1204116507e-10,-.2678274601e-12,
c      .1945349494e-14/
data d7/ .1573501240e-04, .1393650694e-04,-.1111893241e-05,
c      .5965183054e-07,-.1775808250e-08, .2647392778e-10,
c      -.1548306318e-12/

c
snc=2350.
bt(1)=1.386
fr(1)=.68
ro=.204
bt(2)=bt(1)
fr(2)=1.-fr(1)
al(1)=alog(ro)+3.*bt(1)**2
al(2)=alog(ro)+2.*bt(2)**2

```

```

ff=.5
more=0
ncase=0

c
write(*,*)' '
write(*,*)' *** AFCSA/SASM NUCLEAR FALLOUT MODEL ***'
write(*,*)' Version 4.2'
c
Tom Hopkins AFCSA/SASM Oct85

write(*,*)' '
c
write(*,*)' This program computes the x-y coordinates of a user-'
c
write(*,*)' defined dose (rate) contour,'
c
write(*,*)' or dose (rate) at specified coordinates.'
c
write(*,*)' The activity-size distribution is DELFIC-default.'
c
write(*,*)' The bomb fission fraction is set at 50% .'
c
write(*,*)' The source normalization is 2350 (R/hr)/(mi2/kt).'
c
write(*,*)' '
write(*,*(a\))' Enter weapon yield in megatons: '
read(*,*)ym
write(*,*)' '
write(*,*(a\))' Enter wind speed in miles per hour: '
read(*,*)vx
write(*,*)' '
write(*,*(a\))' Enter wind shear in inverse hours: '
read(*,*)shry
write(*,*)' '
write(*,*)' Enter 0 for single burst case,'
write(*,*(a\))' or 1 for multi-burst case: '
read(*,*)mb
if(mb.eq.1)then
write(*,*)' '
write(*,*(a\))' Enter the number of bursts: '
read(*,*)nb
write(*,*(a\))' Enter the burst line length in miles: '
read(*,*)bl
endif

c
write(*,*)' '
write(*,*)' Enter 0 for contour calculation,'
write(*,*(a\))' or 1 for point calculation: '
read(*,*)ncon

c
write(*,*)' '
98 if(ncon.eq.0)then
write(*,*(a\))' Enter desired contour level: '
read(*,*)drc
else
write(*,*)' Enter coordinates of evaluation point.'
write(*,*(a\))' downwind miles: '

```

```

        read(*,*)xpt
        write(*,'(a\)' )'      crosswind miles: '
        read(*,*)ypt
    endif
c
    write(*,*)' '
    write(*,*)' Enter 0 for dose rate,'
    write(*,*)'      1 for infinite time dose,'
    write(*,'(a\)' )'      or 2 for finite time dose: '
    read(*,*)inf
c
        if(inf.eq.2)then
            write(*,*)' '
            write(*,'(a\)' )' Enter arrival time of receiver (hours): '
            read(*,*)tin
            write(*,'(a\)' )' Enter departure time of receiver (hours): '
            read(*,*)tout
        endif
c
        if(more.eq.0)then
            write(*,*)' '
            write(*,*)' Enter 1 to save output on a disk file,'
            write(*,'(a\)' )'      or 0 for screen output only: '
            read(*,*)ndisk
            if(ndisk.eq.1)then
                write(*,'(a\)' )' Enter the output file name: '
                read(*,43)fname
                43      format(a15)
                open(44,file=fname,status='new')
            endif
        endif
c
        write(*,*)' '
        if(ndisk.eq.1)write(44,*)' '
        write(*,*)'      SUMMARY'
        if(ndisk.eq.1)write(44,*)'      SUMMARY'
        write(*,*)' '
        if(ndisk.eq.1)write(44,*)' '
        if(ncon.eq.0)then
            write(*,*)'      Contour Position Calculation'
            if(ndisk.eq.1)write(44,*)'      Contour Position Calculation'
        endif
        if(ncon.eq.1.and.inf.eq.0)then
            write(*,*)'      Point Calculation of Dose Rate'
            if(ndisk.eq.1)write(44,*)'      Point Calculation of Dose Rate'
        endif
        if(ncon.eq.1.and.inf.eq.1)then
            write(*,*)'      Point Calculation of Infinite Time Dose'

```

```

        if(ndisk.eq.1)
c write(44,*)' Point Calculation of Infinite Time Dose'
        endif
        if(ncon.eq.1.and.inf.eq.2)then
            write(*,*)' Point Calculation of Finite Time Dose'
            write(*,7)' Receiver Arrival Time :',tin
            write(*,7)' Receiver Departure Time :',tout
7 format(a30,e9.3)
            if(ndisk.eq.1)then
                write(44,*)' Point Calculation of Finite Time Dose'
                write(44,7)' Receiver Arrival Time :',tin
                write(44,7)' Receiver Departure Time :',tout
            endif
        endif
        if(ncon.eq.0.and.inf.eq.2)then
            write(*,*)' '
            write(*,*)' Finite Time Dose Contour'
            write(*,7)' Receiver Arrival Time :',tin
            write(*,7)' Receiver Departure Time :',tout
            if(ndisk.eq.1)then
                write(44,*)' '
                write(44,*)' Finite Time Dose Contour'
                write(44,7)' Receiver Arrival Time :',tin
                write(44,7)' Receiver Departure Time :',tout
            endif
        endif
        write(*,*)' '
        write(*,1)' Weapon Yield = ',ym,' megatons'
        write(*,2)' Wind Speed = ',vx,' mph'
        write(*,3)' Wind Shear = ',shry,' per hour'
        if(ndisk.eq.1)then
            write(44,*)' '
            write(44,1)' Weapon Yield = ',ym,' megatons'
            write(44,2)' Wind Speed = ',vx,' mph'
            write(44,3)' Wind Shear ',shry,' per hour'
        endif
        if(inf.eq.0.and.ncon.eq.0)then
            write(*,4)' Dose Rate = ',drc,' r/hr'
            if(ndisk.eq.1)write(44,4)' Dose Rate = ',drc,' r/hr'
        endif
        if(inf.eq.1.and.ncon.eq.0)then
            write(*,4)' Infinite Time Dose = ',drc,' r'
            if(ndisk.eq.1)write(44,4)' Infinite Time Dose = ',drc,' r'
        endif
        if(inf.eq.2.and.ncon.eq.0)then
            write(*,4)' Finite Time Dose = ',drc,' r'
            if(ndisk.eq.1)write(44,4)' Finite Time Dose = ',drc,' r'
        endif
    endif

```

```

write(*,*)' '
if(ndisk.eq.1)write(44,*)' '
if(mb.eq.0)then
write(*,*)' Single Burst Case '
if(ndisk.eq.1)write(44,*)' Single Burst Case'
else
write(*,*)' Multiple Burst Case '
write(*,5)nb,' bursts along a line ',bl,' miles long'
if(ndisk.eq.1)then
write(44,*)' Multiple Burst Case'
write(44,5)nb,' bursts along a line ',bl,' miles long'
endif
endif
endif
c
1 format(a25,f10.3,a10)
2 format(a25,f10.0,a10)
3 format(a25,f10.1,a10)
4 format(a25,f10.0,a10)
5 format(i10,a21,f8.1,a11)
c
if(more.eq.1)go to 12
c
c yield-dependent constants
c hc:kft, hk:km, sigh:mi, sigo:mi, tc:hrs
c
hc=44.+6.1*log(ym)-.205*(log(ym)+2.42)*abs(log(ym)+2.42)
hk=hc/3.281
sigh=.18*hc/5.28
sigo=exp(.7+log(ym)/3.-3.25/(4.+(log(ym)+5.4)**2))
tc=1.0573*(12.*hc/60.-2.5*hc*hc/3600.)*(1.-.5*exp(-hc*hc/625.))
c
c laurent series constants from polynomial fits
c
do 11 i=1,7
c(1)=c(1)+d1(i)*hk**(i-1)
c(2)=c(2)+d2(i)*hk**(i-1)
c(3)=c(3)+d3(i)*hk**(i-1)
c(4)=c(4)+d4(i)*hk**(i-1)
c(5)=c(5)+d5(i)*hk**(i-1)
c(6)=c(6)+d6(i)*hk**(i-1)
c(7)=c(7)+d7(i)*hk**(i-1)
11 continue
c
if(ym.lt.1.5)ff=1.-ym/3.
c
12 aa=snc*ff*ym*1000./vx
c
if(ncon.eq.0)then

```



```

        call contur(aa)
        write(*,*)' '
        write(*,*)' Enter 1 to generate another contour,'
        write(*,*(a\))'      or 0 to stop this run: '
        read(*,*)more
        if(more.eq.1)then
            go to 98
        endif
    else
        ta=xpt/vx
        if(inf.eq.2)then
            if(ta.ge.tin.and.ta.lt.tout)tin=ta
            if(ta.ge.tout)then
                drxy=0.
                write(*,*)' Receiver departed before fallout.'
                if(ndisk.eq.1)write(44,*)' Receiver departed before fallout.'
                go to 8
            endif
        endif
        drxy=aa*fxy(xpt,ypt)*g(ta)
8      write(*,6)'xpt = ',xpt,' ypt = ',ypt,' toa = ',ta,' D = ',drxy
6      format(a8,e8.3,a8,e8.3,a8,e8.3,a8,e8.3)
        if(ndisk.eq.1)
c      write(44,6)' xpt = ',xpt,' ypt = ',ypt,' toa = ',ta,' D = ',
c      drxy
        write(*,*)' '
        write(*,*)' Enter 1 to perform another point calculation,'
        write(*,*(a\))'      or 0 to stop this run: '
        read(*,*)more
        if(more.eq.1)then
            go to 98
        endif
    endif
c
c      if(ndisk.eq.1)then
c          write(*,*)' '
c          write(*,*)' Output written to disk file name : ',fname
45      format(a40,a15)
c          close(44)
c      endif
c
c      999 stop
c      end
c
c      *****
c
c      subroutine contur(aa)
c      character gtalk*3, ncpr(10)*1, nc(3)*1

```

```

equivalence (gtalk,nc(1))
common /xy/ tmax,xmax,drmax,xup,xdn,du2,dd2,nocon
common /v/ vx,shry
common /c/ drc
common /inf/ inf,tin,tout
common /m/ mb,nb,bl
common /nd/ ndisk,ncon,ncase

c
data ncpr/'1','2','3','4','5','6','7','8','9','0'/
data nc/'@',' ',' ',' '/
ncase=ncase+1
call search(aa)

c
if(nocon.eq.1)go to 999

c
c Divide the hotline into 10 segments between xup & xdn.
c Compute coordinates of iso-drc contour.
c
y=0.
if(mb.eq.1)y=bl/2.
x=xup
yold=y
t=x/vx
deltax=(xdn-xup)/10.
write(*,*)' '
ms=ncase/10.
ls=mod(ncase,10)
if(ls.eq.0)ls=10
if(ms.eq.0)then
nc(2)=ncpr(ls)
else
nc(2)=ncpr(ms)
nc(3)=ncpr(ls)
endif
if(ndisk.eq.1)write(44,16)gtalk
16 format(a3)
write(*,*)'          x          y          -y          toa'
write(*,13)xup,y,-y,xup/vx
13 format(4e12.3)
if(ndisk.eq.1)then
write(44,*)'          x          y          -y          toa'
write(44,13)xup,y,-y,xup/vx
endif
dy=.5
if(mb.eq.0)dy=.1

c
do 12 i=1,9
x=xup+i*deltax

```

```

      t=x/vx
c
      nfl=0
      y=yold
14    drxy=aa*g(t)*fxy(x,y)
c
      if(drxy.ge.drc)then
        if(nfl.eq.2)go to 69
        y=y+dy
        nfl=1
        go to 14
      elseif(drxy.lt.drc)then
        if(nfl.eq.1)go to 69
        y=y-dy
        nfl=2
        go to 14
      endif
c
69    yold=y
      write(*,13)x,y,-y,x/vx
      if(ndisk.eq.1)write(44,13)x,y,-y,x/vx
c
12    continue
      write(*,13)xdn,0.,0.,xdn/vx
      if(ndisk.eq.1)write(44,13)xdn,0.,0.,xdn/vx
c
999  return
      end
c
c *****
c
      subroutine search(aa)
      common /xy/ tmax,xmax,drmax,xup,xdn,du2,dd2,nocon
      common /v/ vx,shry
      common /c/ drc
      common /inf/ inf,tin,tout
      common /m/ mb,nb,bl
      common /nd/ ndisk,ncon,ncase
c
c Find downwind location of max drxy.
c
      gn=1.
      tmax=.05
      ddl=aa*g(tmax)*fxy(vx*tmax,0.)
21    tmax=tmax+gn*.1
      dd2=aa*g(tmax)*fxy(vx*tmax,0.)
      if(dd2.gt.ddl)then
        ddl=dd2

```

```

        gn=gn+1.
        go to 21
    endif
    tmax=tmax-.05
    xmax=vx*tmax
    drmax=aa*g(tmax)*fxy(xmax,0.)
    write(*,*)' '
    if(drmax.lt.drc) then
        write(*,*)' requested drc is too large'
        nocon=1
        write(*,*)' g(tmax) = ',g(tmax)
        write(*,*)' fxy = ',fxy(xmax,0.)
        go to 999
    endif
c
c  find upwind hotline coordinates of drc contour
c
    dul=aa*g(tmax)*fxy(xmax,0.)
    dn=1.
31  tup=tmax-dn*.05
    du2=aa*g(tup)*fxy(vx*tup,0.)
    if(du2.gt.drc)then
        dn=dn+1
        dul=du2
        go to 31
    endif
    tup=tup+.05
    xup=vx*tup
    du2=aa*g(tup)*fxy(xup,0.)
    write(*,*)' Contour intersections with hotline:'
    write(*,32)' upwind ',xup,' miles'
    if(ndisk.eq.1)then
        write(44,*)' '
        write(44,*)' Contour intersections with hotline:'
        write(44,32)' upwind ',xup,' miles'
    endif
32  format(a20,e9.3,a9)
c
c  find downwind coordinates of drc contour
c
    ddl=aa*g(tmax)*fxy(xmax,0.)
    dn=1.
41  tdn=tmax+dn*.5
    dd2=aa*g(tdn)*fxy(vx*tdn,0.)
    if(dd2.gt.drc)then
        dn=dn+1
        ddl=dd2
        go to 41

```

```

endif
tdn=tdn-.5
xdn=vx*tdn
dd2=aa*g(tdn)*fxy(xdn,0.)
write(*,32)' downwind ',xdn,' miles'
if(ndisk.eq.1)
c write(44,32)' downwind ',xdn,' miles'
999 return
end

c
c *****
c
function fxy(x,y)
common /m/ mb,nb,b1
common /v/ vx,shry
common /inf/ inf,tin,tout
t=x/vx
if(t.lt.0.)t=abs(t)
if(t.eq.0.)t=1.e-06
if(mb.eq.0)then
fdum=(y/sigy(t))**2
fdum=exp(-.5*fdum)
fxy=fdum/(sqrt(6.28318)*sigy(t))
elseif(mb.eq.1)then
z1=(y+b1/2.)/sigy(t)
z=z1
cnz1=cn(z)
z2=(y-b1/2.)/sigy(t)
z=z2
cnz2=cn(z)
fxy=(cnz1-cnz2)*nb/b1
endif
if(inf.eq.0)return
if(inf.eq.1.or.inf.eq.2)then
fdux=fxy
if(inf.eq.1)then
tout=1.e+15
tin=t
endif
fxy=fdux*5.*(tin**(-.2)-tout**(-.2))
return
endif
end

c
c *****
c
function g(t)
common /p/ al(2),bt(2),fr(2),ro,c(7)

```

```

tz=.1
if(t.lt.tz) s=tz
if(t.ge.tz) s=t
r=c(1)/(s**5)+c(2)/(s**4)+c(3)/(s**3)
r=r+c(4)/(s*s)+c(5)/s+c(6)+c(7)/sqrt(s)
r=r*1.e+06
a=0.
do 1 l=1,2
p=(alog(r)-al(1))/bt(1)
a=a+fr(1)*exp(-.5*p*p)/(sqrt(6.283)*bt(1)*r)
if(fr(1).eq.1.)go to 2
1 continue
2 drdt=-5.*c(1)/(s**6)-4.*c(2)/(s**5)-3.*c(3)/(s**4)
drdt=drdt-2.*c(4)/(s**3)-c(5)/(s*s)-.5*c(7)/(s**1.5)
drdt=drdt*1.e+06
gdum=a*abs(drdt)
if(t.ge.tz)then
g=gdum
return
endif
g=gdum*t/tz
return
end

```

```

c
c *****
c

```

```

function sigy(t)
common /t/ tc,sigo,sigh
common /v/ vx,shry
ts=t
if(ts.gt.3) ts=3.
tr=1.+8.*ts/tc
sigy=sqrt(sigo**2*tr+(sigh*shry*t)**2)
return
end

```

```

c
c *****
c

```

```

function cn(z)
x=abs(z)
if(x.le..01) x=.01
if(x.ge.5.) x=5.
cc=1.+1.96854*x+.115194*x*x+.000344*x**3+.019527*x**4
cc=.5/(cc**4)
if(z.ge.0.) cn=1.-cc
if(z.lt.0.) cn=cc
return
end

```